

Our algorithm creates inequalities from each constraint, and then creates CNF clauses from those inequalities, by defining which ones are allowed to coexist and which ones are not allowed. From each constraint, $A < B < C$. We create two different inequalities: $X: A < C$, $Y: B < C$ or $Z: C < A$, $K: C < B$. Then we add (X, K) (Y, Z) into the clause to make sure that we pick X and Y or Z and K . Then we run a cross checking between all the existing constraints to make sure that we don't allow conflicted constraints, X, Y to exist by adding clause $(\text{not } X, \text{not } Y)$ (more details are in the code).

Then, it uses a free SAT solver to solve the CNF. Using the solutions from the SAT solver, we determine which inequalities create a valid ordering. Lastly, we use those known comparisons between wizards to place them in order.

We used Pycosat 0.6.3 as our SAT solver because we don't know how to write one on our own.

To install Pycosat:

```
> python3 setup.py install
```

We are running out of time to write the write up. More details are in the code